simulationexampleoutput

2024-03-18

```
# Load required libraries
library(sampling)
library(survey)
```

```
## Loading required package: grid
```

```
## Loading required package: Matrix
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following objects are masked from 'package:sampling':
##
##     cluster, strata
```

```
##
## Attaching package: 'survey'
```

```
## The following object is masked from 'package:graphics':
##
##     dotchart
```

```
library(SDAResources)

# Load data
data(agpop)
# remove missing value of acres92
agpopc <- agpop[complete.cases(agpop$acres92), ]
nrow(agpopc)
```

```
## [1] 3059
```

```
# Sort the data by region for stratification
agpop_sort <- agpopc[order(agpopc$region),]
head(agpop_sort)
```

```
##               county state acres92 acres87 acres82 farms92 farms87 farms82
## 525     ADAIR COUNTY    IA  328970  353365  355269     891     970    1065
## 526     ADAMS COUNTY    IA  239800  243607  254071     643     688     737
## 527 ALLAMAKEE COUNTY    IA  321728  321226  351756    1000    1062    1134
## 528 APPANOOSE COUNTY    IA  238609  244661  236501     827     891     900
## 529   AUDUBON COUNTY    IA  268506  268437  283782     740     851     905
## 530    BENTON COUNTY    IA  427215  419480  441810    1325    1434    1585
##     largef92 largef87 largef82 smallf92 smallf87 smallf82 region
## 525       56       53       42       55       63       48     NC
## 526       38       32       21       40       50       33     NC
## 527       46       37       37       55       61       56     NC
## 528       28       33       23       25       39       36     NC
## 529       45       30       25       60       79       39     NC
## 530       56       43       34      101      105      102     NC
```

```
# Calculate the length of each region
length_region <- c(length(agpop_sort$region[agpop_sort$region == "NC"]),
                   length(agpop_sort$region[agpop_sort$region == "NE"]),
                   length(agpop_sort$region[agpop_sort$region == "S"]),
                   length(agpop_sort$region[agpop_sort$region == "W"]))

# Display the lengths of each region
length_region
```

```
## [1] 1052  213 1376  418
```

```
sum(length_region)
```

```
## [1] 3059
```

```
round(length_region*0.1)
```

```
## [1] 105  21 138  42
```

```
sum(round(length_region*0.1))
```

```
## [1] 306
```

```
n<-c(103,20,136,41)
sum(n)
```

```
## [1] 300
```

```
# (a) Draw a stratified random sample of size 300 according to proportional
# allocation (NC : 103; NE :20; S : 136; W : 41).

index <- c(sample(1:length_region[1], n[1], replace = FALSE),
           sample((length_region[1] + 1):sum(length_region[1:2]), n[2],
                  replace = FALSE),
           sample((sum(length_region[1:2]) + 1):sum(length_region[1:3]), n[3],
```

```
                replace = FALSE),
          sample((sum(length_region[1:3]) + 1):sum(length_region[1:4]), n[4],
                replace = FALSE))
index
```

```
##   [1]  617  620  647   78  961  186  868 1011  661  290  490  515  800  956  505
##  [16]  256  159  744  908  327  547  879  309   67   75  570  700  513  863  573
##  [31]  648  318  546   55  779 1005  781  213   59 1039  215  698   17  203  526
##  [46]   86  952  564  954  433  844  743  144  416  726  668  767  876  443  551
##  [61]  438  154  621  254  559  516  485 1023  632  944  293  703  613  752  893
##  [76]  157    7  807  553  340  158  710  222  445  426  458  301  151  720  985
##  [91]  603   65  481  139  223  297  391  257  586  423  202  886  536 1221 1212
## [106] 1115 1159 1250 1128 1126 1191 1188 1164 1177 1112 1104 1246 1087 1101 1258
## [121] 1077 1106 1144 1990 1280 1501 1760 1806 1506 1469 1477 1996 1433 1538 2561
## [136] 2556 2463 2555 1905 2604 2478 2200 2471 2158 1357 1604 1863 1982 2073 2185
## [151] 2272 1282 1629 2027 1420 1342 2333 1308 2381 1955 1532 2580 1713 2193 1432
## [166] 1320 2325 1343 2560 1514 1879 1517 2630 2443 2029 2052 1556 1373 1823 2210
## [181] 2066 1959 1398 1347 1887 2501 2100 1702 2207 2516 2494 1989 2246 2315 2253
## [196] 2287 2430 2223 2305 2543 1512 2444 1734 2290 2247 2563 1958 2633 2454 1323
## [211] 1904 1674 2307 2255 1774 2110 1447 2401 1527 1938 2529 1311 1633 1757 2539
## [226] 2537 1834 2010 1441 2095 2147 1745 1334 2554 1369 1705 1524 1529 1847 2231
## [241] 2621 1867 1793 1926 1483 1689 2129 2371 2268 1518 1338 1873 1704 1442 1474
## [256] 1332 1403 2617 1748 2755 2643 2869 2721 2750 2715 2653 2990 2685 2724 2853
## [271] 2722 3059 2978 2857 3052 3042 2651 3058 2970 2763 2975 2756 2943 2822 2884
## [286] 3010 2690 2743 2762 2745 2997 2661 2732 2951 3036 2890 2914 2720 2808 2674
```

```
agprop <- agpop_sort[index,]
head(agprop)
```

```
##                   county state acres92 acres87 acres82 farms92 farms87 farms82
## 1422       JASPER COUNTY    MO  281327  275511  272986    1317    1418    1427
## 1425         KNOX COUNTY    MO  268447  292328  276894     587     651     680
## 1452        PERRY COUNTY    MO  209452  221634  225584     897     959    1016
## 602  POTTAWATTAMI COUNTY    IA  542855  543881  555348    1441    1652    1828
## 2378     MARSHALL COUNTY    SD  485748  468868  490427     487     522     526
## 754        SHELBY COUNTY    IL  402212  422071  420047    1305    1431    1599
##      largef92 largef87 largef82 smallf92 smallf87 smallf82 region
## 1422       38       28       32       66       86       73     NC
## 1425       67       62       54       29       27       32     NC
## 1452       26       29       25       42       66       70     NC
## 602        99       83       72      102      147      132     NC
## 2378      171      164      170       18       24       22     NC
## 754        73       61       43       76       93       81     NC
```

```
# (b) Calculate estimated population total \hat t_{str} for acres92
dstr <- svydesign(id = ~1, strata = ~region,
                fpc = c(rep(length_region[1], n[1]),
                        rep(length_region[2], n[2]),
                        rep(length_region[3], n[3]),
                        rep(length_region[4], n[4])),
                data = agprop)
dstr
```

```
## Stratified Independent Sampling design
## svydesign(id = ~1, strata = ~region, fpc = c(rep(length_region[1],
##     n[1]), rep(length_region[2], n[2]), rep(length_region[3],
##     n[3]), rep(length_region[4], n[4])), data = agprop)
```

```
svytotal(~acres92, dstr)
```

```
##             total       SE
## acres92 938137708 45364625
```

```
strtotal <- data.frame(svytotal(~acres92, dstr))[1, 1]
strtotal
```

```
## [1] 938137708
```

```
# (c) Repeat the above for 1000 times, draw a histogram of the 1000 estimates
#of the total you got.
# Record mean and median of the 1000 standard error estimates.

# Define a function for simulation
simu <- function(T, n) {
  # T is the number of simulation runs, n is the vector of allocation

  # Initialize variables
  strtotal <- rep(NA, T)

  # Loop for T simulations
  for (i in 1:T) {
    # Generate sample indices
    # Generate sample indices
    index <- c(sample(1:length_region[1], n[1], replace = FALSE),
               sample((length_region[1] + 1):sum(length_region[1:2]), n[2],
                      replace = FALSE),
               sample((sum(length_region[1:2]) + 1):sum(length_region[1:3]), n[3],
                      replace = FALSE),
               sample((sum(length_region[1:3]) + 1):sum(length_region[1:4]), n[4],
                      replace = FALSE))

    # Sample data
    agprop <- agpop_sort[index,]

    # Create survey design object
    dstr <- svydesign(id = ~1, strata = ~region,
                      fpc = c(rep(length_region[1], n[1]),
                              rep(length_region[2], n[2]),
                              rep(length_region[3], n[3]),
                              rep(length_region[4], n[4])),
                      data = agprop)

    # Calculate total for acres92
    strtotal[i] <- data.frame(svytotal(~acres92, dstr))[1, 1]
  }
```

```r
  # Calculate mean and median of the standard error estimates
  meanstrtotal <- mean(strtotal)
  medianstrtotal <- median(strtotal)

  # Return results
  list(strtotal = strtotal, meanstrtotal = meanstrtotal, medianstrtotal = medianstrtotal)
}

# Define allocation vector

n<-c(103,20,136,41)
# Run the simulation
result <- simu(1000, n)

# Display mean and median of the standard error estimates
result$meanstrtotal
```

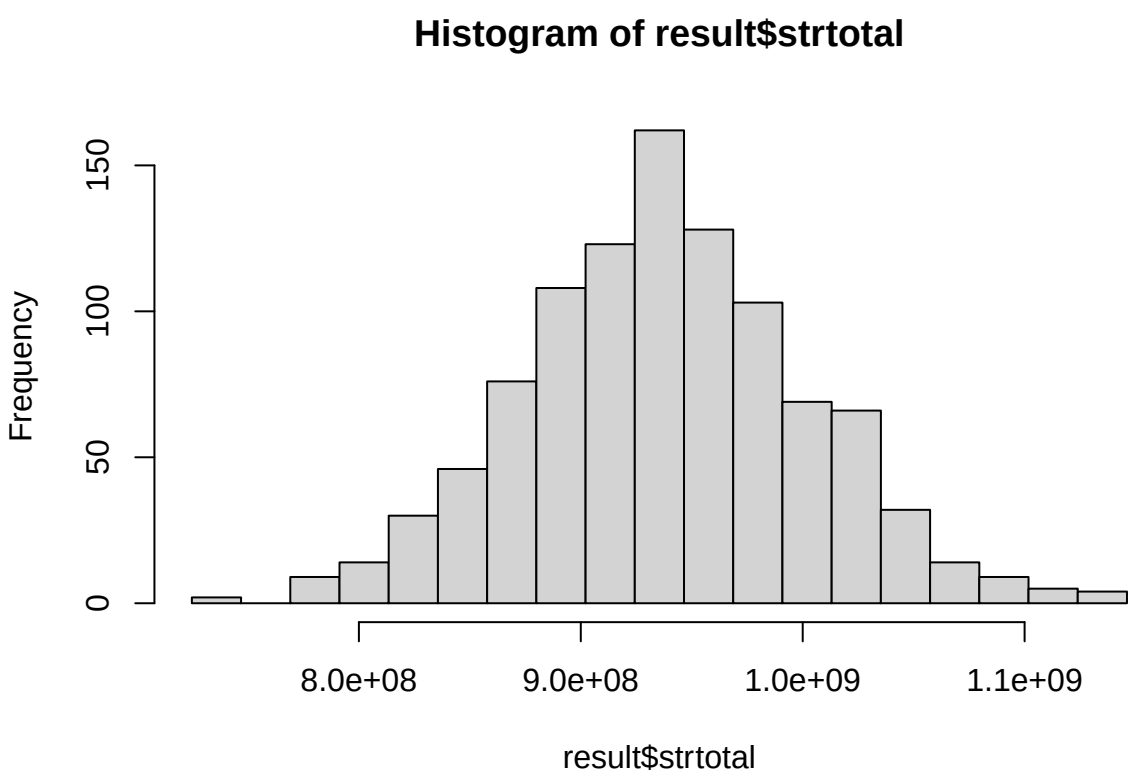
```
## [1] 938355543
```

```r
result$medianstrtotal
```

```
## [1] 937431946
```

```r
# Plot histogram of the standard error estimates
# Specify breaks manually
breaks <- seq(min(result$strtotal), max(result$strtotal), length.out = 20)

# Plot histogram with manual breaks
hist(result$strtotal, breaks = breaks)
```


Histogram of result$strtotal
Frequency
0
50
100
150
8.0e+08
9.0e+08
1.0e+09
1.1e+09
result$strtotal